Bispectrum Estimation Methods

Advanced Digital Signal Processing

Christos Samaras

1 Introduction

This study aims to understand the characteristics of Bispectrum as well as the differences between the various methods of its computation. For this purpose, signal $X[k]$ is constructed as follows

$$X[k] = \sum_{i=1}^{6} cos(\omega_i k + \phi_i) \quad \text{, where k=0, 1, ..., N-1 and N=8192}$$

and $\omega_i = 2\pi\lambda_i$, with $\lambda_1 = 0.12Hz$, $\lambda_2 = 0.30Hz$, $\lambda_4 = 0.19Hz$, $\lambda_5 = 0.17Hz$, $\lambda_3 = \lambda_1 + \lambda_2 = 0.42Hz$ and $\lambda_6 = \lambda_4 + \lambda_5 = 0.36Hz$. Variables ϕ_1, ϕ_2, ϕ_4 and ϕ_5 are independent and uniformly distributed random variables on $[0, 2\pi]$, $\phi_3 = \phi_1 + \phi_2$ and $\phi_6 = \phi_4 + \phi_5$.

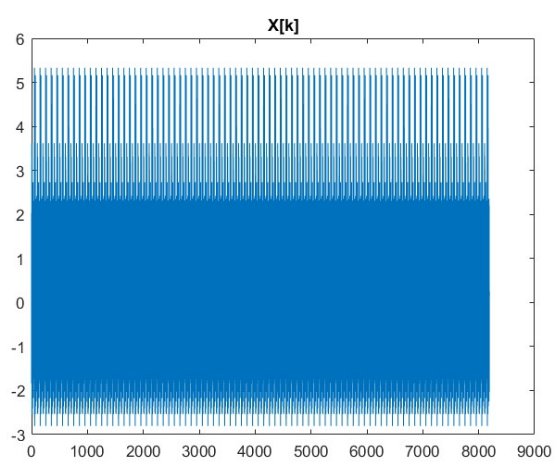

Figure 1: Original signal X[k]

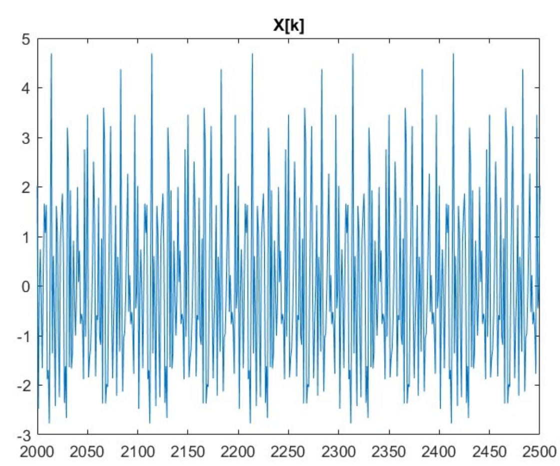

Figure 2: 500 samples of X[k]

2 Power Spectrum

According to Wiener-Khinchin-Einstein theorem, Power Spectrum is equal to Fourier Transform of auto-correlation sequence $m_2^x(\tau)$. Using 128 shiftings for auto-correlation, Power Spectrum is estimated and plotted below.

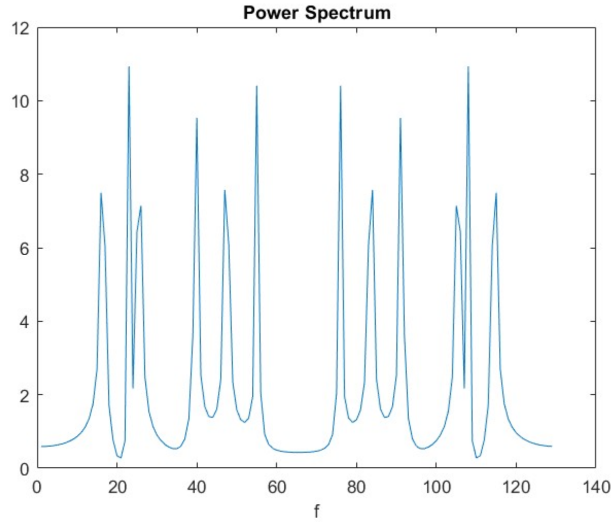

Figure 3: Power Spectrum of X[k]

3 Bispectrum via Indirect vs Direct Method

Bispectrum $C_3^x(f_1, f_2)$ is defined as the 2D Discrete Fourier Transform of 3$^{\text{rd}}$ order cumulants $c_3^x(\tau_1, \tau_2)$

$$C_3^x(f_1, f_2) = \sum_{\tau_1=-\infty}^{+\infty} \sum_{\tau_2=-\infty}^{+\infty} c_3^x(\tau_1, \tau_2) e^{-j(f_1\tau_1 + f_2\tau_2)}$$

First, signal is divided into $K = 32$ segments of $M = 256$ samples. Now, bispectrum is estimated only in primary area via the indirect (using "bispeci" function from HOSA Toolbox) and direct method (using "bispecd" function from HOSA Toolbox). Bispectrum is estimated with indirect method using both Rectangular Window and Parzen Window. For rectangular window, "bispeci" function is slightly changed("bispeci_rect.m" file). Considering that FFT on discrete signals is similar to applying rectangular window on its edges, the part of "bispeci" code that applies window is just erased. Bispectrum plots are displayed right after.

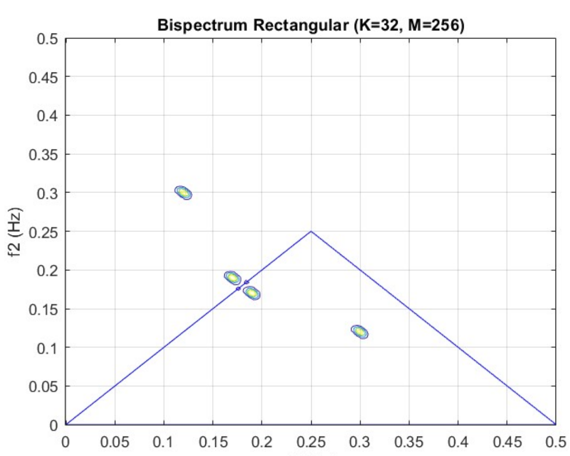

Figure 4: Bispectrum with Rectangular Window

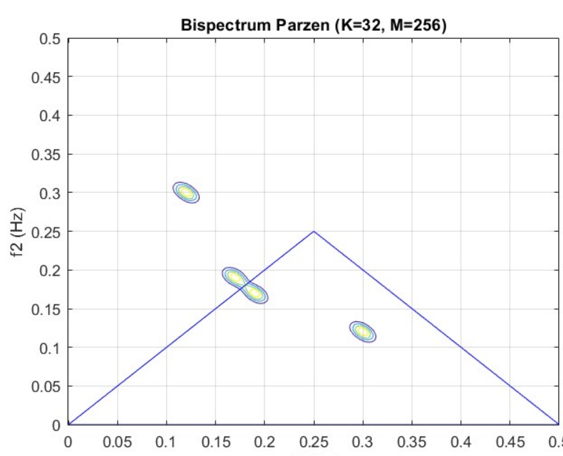

Figure 5: Bispectrum with Parzen Window

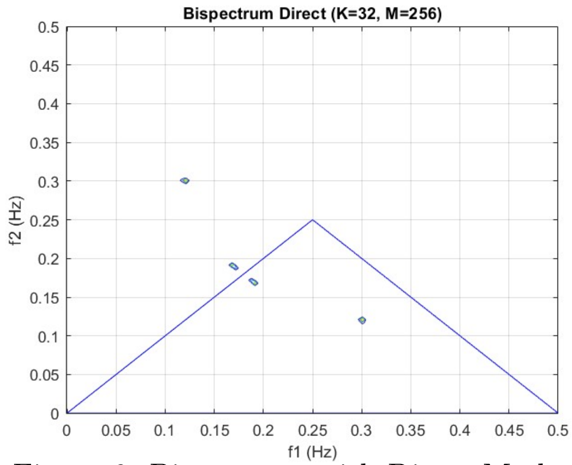

Figure 6: Bispectrum with Direct Method

Observing these plots, it seems that indirect method using rectangular window leads to sharp edges in bispectrum domain. In contrast, using Parzen Window, bispectrum plot is more smooth. Thus, rectangular window probably causes spectral leakage. This phenomenon is due to windows' shape. Rectangular window has sharp edges in contrast with Parzen Window which fades more smoothly. Here are their shapes.

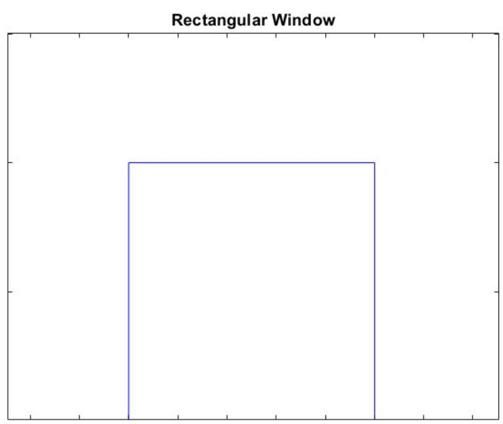

Figure 7: Rectangular Window

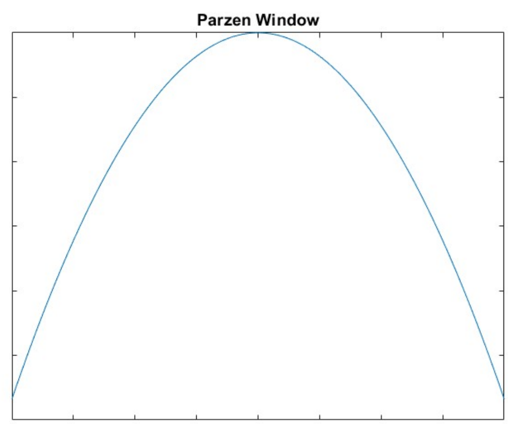

Figure 8: Parzen Window

When it comes to direct method, it can be said that bispectrum content is more concentrated around frequencies that join quadratic phase coupling. This is due to the direct use of Fast Fourier Transform leading to more accurate localization and imaging compared to indirect method.

4 Relation between Spectrum and Bispectrum

From signal construction, it is known that $\lambda_3 = \lambda_1 + \lambda_2$, $\phi_3 = \phi_1 + \phi_2$ and $\lambda_6 = \lambda_4 + \lambda_5$, $\phi_6 = \phi_4 + \phi_5$. Looking Power Spectrum plot, 6 peaks are detected (rest 6 are symmetric). X-axe values of these peaks are $16, 23, 26, 40, 47, 55$ respectively. Dividing them by max shiftings $L_2 = 128$, the results are $0.125\,(\approx \lambda_1 = 0.12Hz)$, $0.179\,(\approx \lambda_5 = 0.17Hz)$, $0.203\,(\approx \lambda_4 = 0.19Hz)$, $0.312\,(\approx \lambda_2 = 0.30Hz)$, $0.367\,(\approx \lambda_6 = 0.36Hz)$, $0.429\,(\approx \lambda_3 = 0.42Hz)$ respectively. Namely, Spectrum peaks are approaching frequencies that join Quadratic Phase Coupling. As for Bispectrum, there is content only around QPC frequencies making it an ideal tool for studying nonlinear phenomena.

5 Process repeat with different parameters

Now, Bispectrum is re-estimated with different parameters K and M for all methods. Specifically, it is estimated firstly for $K = 16$ and $M = 512$ and after for $K = 64$ and $M = 128$. Results are shown in figures 9, 10, 11 for the first occasion and in figures 12, 13, 14 for the second occasion.

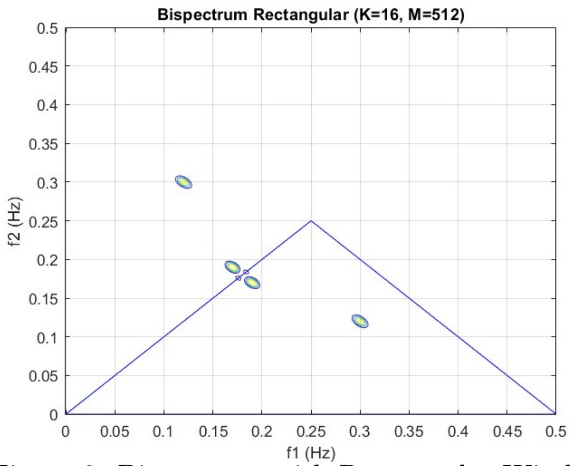

Figure 9: Bispectrum with Rectangular Window

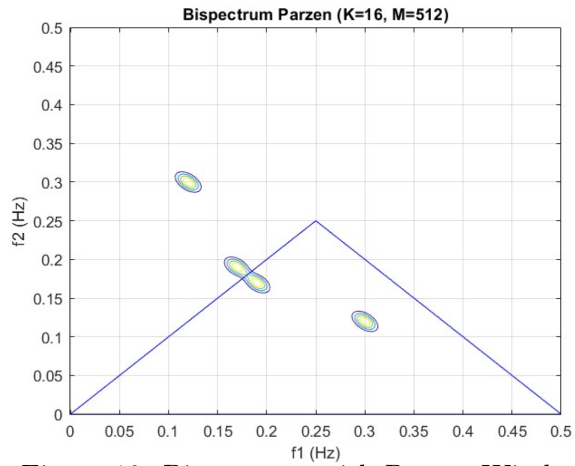

Figure 10: Bispectrum with Parzen Window

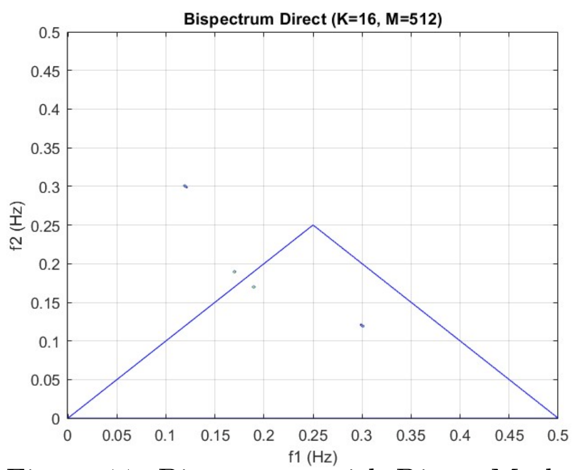

Figure 11: Bispectrum with Direct Method

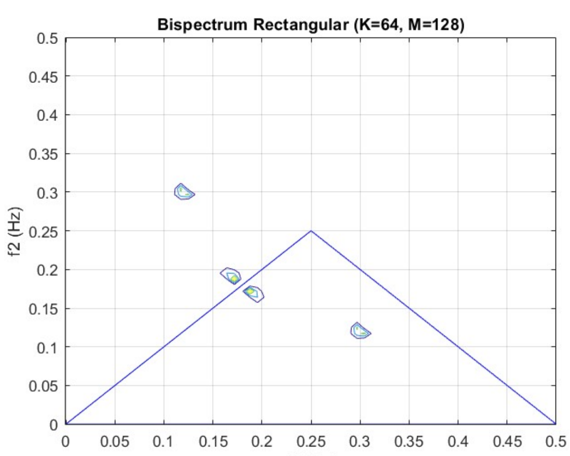

Figure 12: Bispectrum with Rectangular Window

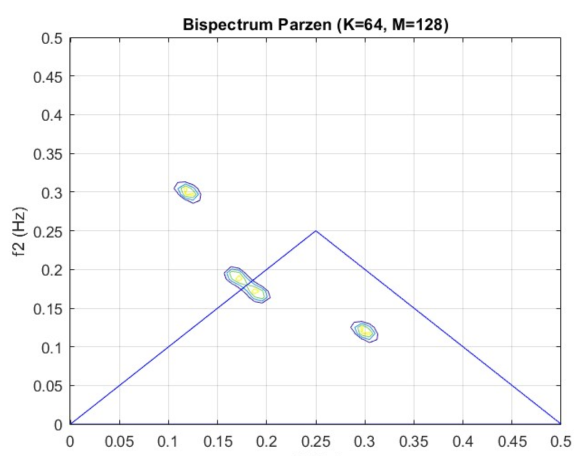

Figure 13: Bispectrum with Parzen Window

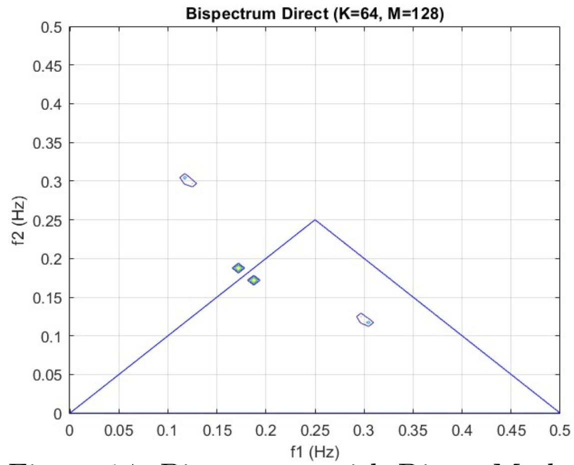

Figure 14: Bispectrum with Direct Method

Comparing plots for the various K and M values, it can be said that increase of samples M and so decrease of segments K increases resolution and precision. Especially for M=512, direct method has extremely shrunken peaks, pointing with great accuracy QPC frequencies. In contrast, increase of segments K and thus decrease of samples M decreases precision leading to appearance of edges which indicates spectral leakage.

Finally, 50 realizations of X[k] are created and the mean spectrum and bispectrum are estimated. For bispectrum, corresponding values of each realization are added and finally divided by number of realizations. As previous, in spectrum, 6 peaks are detected (rest 6 are symmetric). X-axe values of these peaks are $16, 23, 25, 39, 47, 55$ respectively. Dividing them by max shiftings $L = 128$, the results are $0.125\,(\approx \lambda_1 = 0.12Hz)$, $0.179\,(\approx \lambda_5 = 0.17Hz)$, $0.195\,(\approx \lambda_4 = 0.19Hz)$, $0.304\,(\approx \lambda_2 = 0.30Hz)$, $0.367\,(\approx \lambda_6 = 0.36Hz)$, $0.429\,(\approx \lambda_3 = 0.42Hz)$ respectively. As seen, after 50 realizations, the original QPC frequencies are approached with greater accuracy compared to 1 realization.

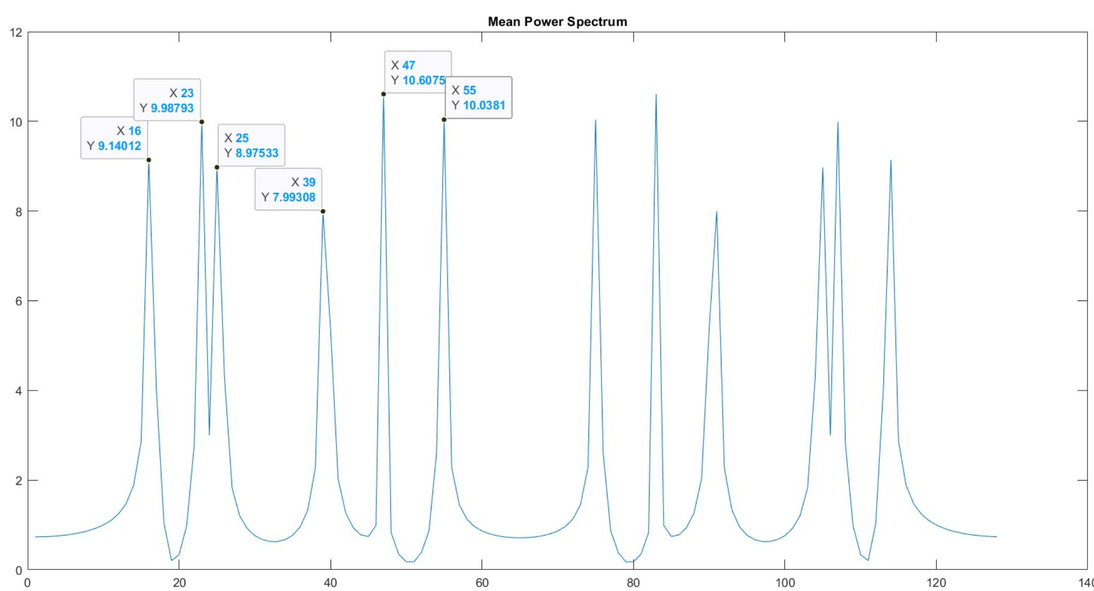

Figure 15: Peaks of Mean Power Spectrum

As for bispectrum, mean bispectrum is estimated for indirect method using Parzen and Rectangular window in order to verify if accuracy of this method is improved. Peaks are shrunken compared to the corresponding peaks of 1 realization, so the accuracy has improved. Still, however, the direct method is more detailed. Probably, this accuracy difference between 2 methods is due to the fact that indirect method applies 2D-FFT to 3$^{\text{rd}}$-order cumulants of signal in contrast with direct method which applies FFT directly to the signal.

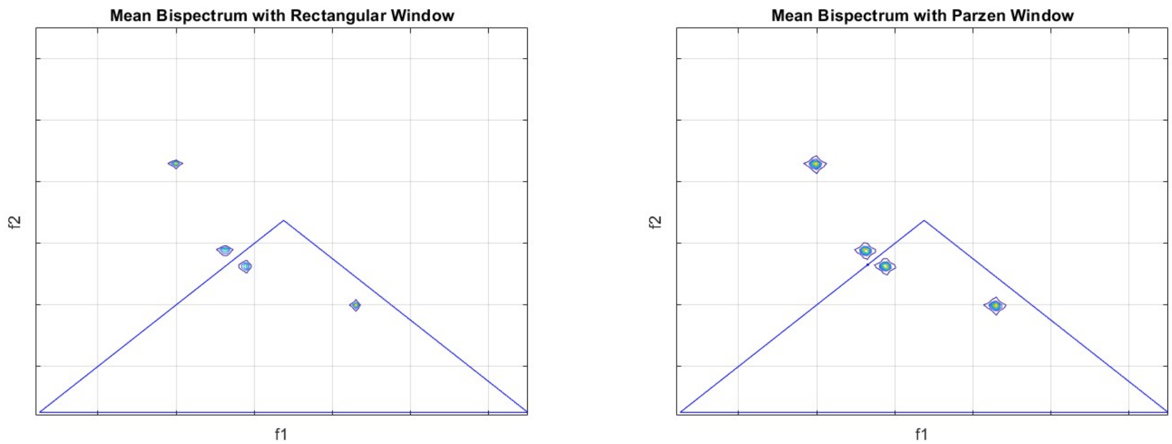

Figure 16: Mean bispectrum using Rectangular and Parzen Window

6 MATLAB code

```
%% ex2.1 - Signal construction

phi1=rand*2*pi; phi2=rand*2*pi; phi4=rand*2*pi; phi5=rand*2*pi;
phi3=phi1+phi2; phi6=phi4+phi5;
phi=[phi1, phi2, phi3, phi4, phi5, phi6];
w=[2*pi*0.12, 2*pi*0.3, 2*pi*0.42, 2*pi*0.19, 2*pi*0.17, 2*pi*0.36];

N = 8192; K = 0:N-1;
X = zeros(size(K));
for kindx = 1:length(K)
    X(kindx) = sum(cos(w .* K(kindx) + phi));
end
figure;
plot(k, X)
title('X[k]')
figure;
plot(K, X)
xlim([2000 2500])
title('X[k]')
fprintf('-----------------------------------------------------------------')
%% ex2.2 - Autocorrelation and Power Spectrum
acf = autocorr(X, 'NumLags', 128);
Pxx = fft(acf);
figure;
plot(abs(Pxx))
xlabel('f');
title('Power Spectrum');
fprintf('-----------------------------------------------------------------')
%% ex2.3 & ex2.7 - Bispectrum Estimation
nlag = 64; flag = 'unbiased'; overlap = 0; nsampValues = [256, 512, 128];
K = [32, 16, 64]; methods = {'indirect-rect', 'indirect-parzen', 'direct'};

for j = 1:3  % Loop over nsamp and nfft values
    nsamp = nsampValues(j);
    nfft = K(j);

    % Task a1 - Rectangular Window
    figure;
    bispeci_rect(X, nlag, nsamp, overlap, flag, nfft);
    plotBispectrum(sprintf('Bispectrum Rectangular (K=%d, M=%d)', nfft, nsamp));

    % Task a2 - Parzen Window
    figure;
    bispeci(X, nlag, nsamp, overlap, flag, nfft, 0); % Parzen window
    plotBispectrum(sprintf('Bispectrum Parzen (K=%d, M=%d)', nfft, nsamp));

    % Direct Method
    figure;
    bispecd(X, nsamp, 1, nsamp, 50); %J=0 nfft=nsamp
    plotBispectrum(sprintf('Bispectrum Direct (K=%d, M=%d)', nfft, nsamp));
    fprintf('-----------------------------------------------------------------')
end

% ex2.7 task b - 50 realizations
nsamp=256; nfft=256; wind=1; overlap=50;
NumLags=128;
C2values = zeros(NumLags, 50);
```

```matlab

sumC3rect = zeros(nfft, nfft);
sumC3parzen = zeros(nfft, nfft);

for j = 1:50
    phi1=rand*2*pi; phi2=rand*2*pi; phi3=phi1+phi2; phi4=rand*2*pi;
    phi5=rand*2*pi; phi6=phi4+phi5;
    p = [phi1, phi2, phi3, phi4, phi5, phi6];
    w=[2*pi*0.12, 2*pi*0.3, 2*pi*0.42, 2*pi*0.19, 2*pi*0.17, 2*pi*0.36];

    X = zeros(size(k));
    for idx_k = 1:length(k)
        sumX_k = 0;
        for i = 1:length(w)
            sumX_k = sumX_k + cos(w(i) * k(idx_k) + p(i));
        end
        X(idx_k) = sumX_k;
    end

    % Compute bispectrum using rectangular window
    [C3rect, ~] = bispeci_rect(X, nsamp, nfft, wind, overlap);
    sumC3rect = sumC3rect + C3rect;

    % Compute bispectrum using Parzen window
    [C3parzen, ~] = bispeci(X, nsamp, nfft, wind, overlap);
    sumC3parzen = sumC3parzen + C3parzen;

    % Compute power spectrum (C2)
    acf = autocorr(X, 'NumLags', 128);
    C2 = abs(fft(acf,NumLags));
    C2values(:, j) = C2;
end

% mean bispectrum
meanC3rect = sumC3rect / 50;
meanC3parzen = sumC3parzen / 50;


figure;
contour(abs(meanC3rect));
hold on;
grid on;
plot([125, 187.5], [125, 187.5], 'b', 'LineWidth', 0.15);       % deterimine
plot([187.5, 250], 250 - [62.5, 125], 'b', 'LineWidth', 0.15);  % primary
plot([125, 250], [125, 125], 'b', 'LineWidth', 0.15);           % area
xlim([124 250]);
ylim([124 250]);
xticklabels({});
yticklabels({});
xlabel('f1');
ylabel('f2');
title('Mean Bispectrum with Rectangular Window');
hold off;

figure;
contour(abs(meanC3parzen));
hold on;
grid on;
plot([125, 187.5], [125, 187.5], 'b', 'LineWidth', 0.15);       % deterimine
plot([187.5, 250], 250 - [62.5, 125], 'b', 'LineWidth', 0.15);  % primary
```

```
plot([125, 250], [125, 125], 'b', 'LineWidth', 0.15);            % area
xlim([124 250]);
ylim([124 250]);
xticklabels({});
yticklabels({});
xlabel('f1');
ylabel('f2');
title('Mean Bispectrum with Parzen Window');
hold off;

% mean power spectrum
meanC2 = mean(C2values, 2);
figure;
plot(meanC2);
title('Mean Power Spectrum');

function plotBispectrum(titleStr)
    hold on;
    plot([0,0.25],[0,0.25],'b');           % determine
    plot([0.25,0.5],1/2-[0.25,0.5],'b');  % primary
    plot([0,0.5],[0,0],'b');              % area
    hold off;
    xlim([-0.001 0.5]);
    ylim([-0.001 0.5]);
    xlabel('f1 (Hz)');
    ylabel('f2 (Hz)');
    title(titleStr);
end
```